

"Express Mail" mailing label number EV 314841675 US

Date of Deposit October 13, 2003

Case No. 9683/160

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Dai KAMIYA
Kazuhiro YAMADA
Takashi KONDO

Serial No.: To Be Assigned

Filing Date: October 13, 2003

For: COMMUNICATION DEVICE, PROGRAM
AND RECORDING MEDIA

)
)
)
) Examiner: To Be Assigned
)
) Group Art Unit No.: To Be Assigned
)
)

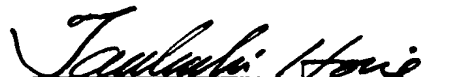
SUBMISSION OF CERTIFIED COPY OF FOREIGN PRIORITY DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants submit herewith a certified copy of Japanese Patent Application No. JP 2002-316635 filed October 30, 2002, to which the above-identified United States Patent Application claims the right of foreign priority under 35 U.S.C. § 119.

Respectfully submitted,


Tadashi Horie
Registration No. 40,437
Agent for Applicants

BRINKS HOFER GILSON & LIONE
P.O. BOX 10395
CHICAGO, ILLINOIS 60610
(312) 321-4200

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年10月30日

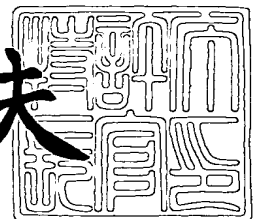
出願番号
Application Number: 特願2002-316635
[ST. 10/C]: [JP2002-316635]

出願人
Applicant(s): 株式会社エヌ・ティ・ティ・ドコモ

2003年 9月22日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3077899

【書類名】 特許願

【整理番号】 DCMH140447

【提出日】 平成14年10月30日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/00
H04Q 7/28

【発明の名称】 通信装置、プログラムおよび記録媒体

【請求項の数】 12

【発明者】

【住所又は居所】 東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ
・ ティ ・ ティ ・ ドコモ内

【氏名】 神谷 大

【発明者】

【住所又は居所】 東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ
・ ティ ・ ティ ・ ドコモ内

【氏名】 山田 和宏

【発明者】

【住所又は居所】 東京都千代田区永田町二丁目 1 1 番 1 号 株式会社エヌ
・ ティ ・ ティ ・ ドコモ内

【氏名】 近藤 隆

【特許出願人】

【識別番号】 392026693

【氏名又は名称】 株式会社エヌ・ティ・ティ・ドコモ

【代理人】

【識別番号】 100098084

【弁理士】

【氏名又は名称】 川▲崎▼ 研二

【選任した代理人】

【識別番号】 100111763

【弁理士】

【氏名又は名称】 松本 隆

【手数料の表示】

【予納台帳番号】 038265

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 通信装置、プログラムおよび記録媒体

【特許請求の範囲】

【請求項 1】 データを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、
該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記特定手段に特定されたデータの使用について、前記オブジェクト生成手段により生成されたオブジェクトの使用のみを前記実行手段に許可するアクセス制御手段と

を有することを特徴とする通信装置。

【請求項 2】 前記選択手段は、前記特定手段に特定されたデータを内包するオブジェクトの型を、前記取得手段により取得された前記識別情報によって示される確率が予め定められた基準確率より低い場合には前記完全カプセル化型とし、前記基準確率以上の場合には前記非完全カプセル化型とする

ことを特徴とする請求項 1 に記載の通信装置。

【請求項 3】 データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、

該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段と

を有することを特徴とする通信装置。

【請求項 4】 前記使用オブジェクト制御手段は、前記取得手段により取得された前記識別情報によって示される確率が、予め定められた第 1 の基準確率より高い場合には、前記完全カプセル化型及び前記非完全カプセル化型のオブジェクトの使用を前記実行手段に許可し、

前記確率が、前記第 1 の基準確率以下であり、かつ予め定められた第 2 の基準確率より高い場合には、前記完全カプセル型のオブジェクトの使用を前記実行手段に許可し、

前記確率が、前記第 2 の基準確率以下である場合には、いずれのオブジェクトの使用も前記実行手段に許可しないことを特徴とする請求項 3 に記載の通信装置。

【請求項 5】 データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、
該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取

得する取得手段と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、

前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、

前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段と

を有することを特徴とする通信装置。

【請求項 6】 前記オブジェクト生成手段により生成された前記オブジェクトを前記プログラムの実行終了に応じて削除する削除手段をさらに有することを特徴とする請求項 1、3 または 5 に記載の通信装置。

【請求項 7】 パケット通信機能を有する携帯電話機であることを特徴とする請求項 1、3 または 5 に記載の通信装置。

【請求項 8】 前記プログラムは、Javaプログラミング言語で記述されたプログラムであって、

前記プログラムを実行するためのJava実行環境を有していることを特徴とする請求項 1、3 または 5 に記載の通信装置。

【請求項 9】 コンピュータに、
データを記憶する記憶機能と、
該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、
該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、

前記特定機能に特定されたデータの使用について、前記オブジェクト生成機能により生成されたオブジェクトの使用のみを前記実行機能に許可するアクセス制御機能と

を実現させるためのプログラム。

【請求項 10】 コンピュータに、

データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、
該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択され

た型のオブジェクトを生成するオブジェクト生成機能と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能と

を実現させるためのプログラム。

【請求項 11】 コンピュータに、

データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、

該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、
該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、

前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、

前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、

内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、

前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、

前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能と

を実現させるためのプログラム。

【請求項 12】 請求項 9 乃至 11 に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、通信装置に格納されたデータに対するセキュリティを確保するための技術に関する。

【0002】

【従来の技術】

近年、パケット通信機能を有する携帯電話機を用いて、インターネットに接続されているサーバから様々なプログラムをダウンロードすることが一般化しつつある。

【0003】

ところで、世界中の様々な人々は、インターネットにおいて、自由に情報の公開やプログラムの提供を行うことができる。インターネットは、このような利点を有する反面、欠点もある。例えば、悪意の有る個人や団体によって、通信装置内に記憶されているデータを密かに盗み出すプログラムが、インターネットを介して提供されてしまう場合がある。あるいは、悪意は無いものの通信装置において動作させると不具合を引き起こしてしまうプログラムが、インターネットを介して提供されてしまうことがある。

したがって、インターネットを介して提供されたプログラムに対して、通信装置の内部および外部のリソースを何ら制限することなくアクセスできるようにしてしまうと、例えば、通信装置内に記憶されているユーザの電話番号やメールアドレス、銀行口座番号などが勝手に読み出され、通信装置の外へ流出してしまうといった事態が生じ得る。

【0004】

このため、例えば、Java（登録商標）言語で記述されたプログラムを実行することが可能な携帯電話機は、インターネットを介してダウンロードされたJavaプログラムを実行するときにアクセスできるリソースを、このプログラムのダウンロード元のサーバと、このプログラムに対して割り当てられた記憶領域のみに制限し、それ以外のユーザの電話番号やメールアドレス、電話帳データ等の個人情報等のリソースには一切アクセスすることができないようにしていた。一方、携帯電話機は、携帯電話機に記憶されている個人情報を、ネイティブプログラムのみで操作するようにして、個人情報のセキュリティを確保していた（例えば、非

特許文献 1 参照)。なお、ここでは、携帯電話機の販売以前にそのメモリに書き込まれるプログラムをネイティブプログラムと呼ぶ。

【0 0 0 5】

【非特許文献 1】

i アプリコンテンツ開発ガイド for 5 0 4 i 詳細編 インターネット<URL : http://www.nttdocomo.co.jp/p_s/imode/java/>

【0 0 0 6】

【発明が解決しようとする課題】

しかしながら、上述したアクセス制限の仕組みは、携帯電話機におけるセキュリティを確保する上で一定の効果を奏するものの、ダウンロードされたプログラムに対して様々な動作制限を課すことになり、プログラムの多様化を阻む要因の一つになっていた。

本発明は、以上説明した事情に鑑みてなされたものであり、携帯電話機等の通信装置に記憶されているデータのセキュリティを確保しつつ、当該データを使用する様々なプログラムを提供できるようにする技術を提供することを目的としている。

【0 0 0 7】

【課題を解決するための手段】

上記課題を解決するために、この発明は、データを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを

生成するオブジェクト生成手段と、前記特定手段に特定されたデータの使用について、前記オブジェクト生成手段により生成されたオブジェクトの使用のみを前記実行手段に許可するアクセス制御手段とを有することを特徴とする通信装置を提供する。

【0008】

また、この発明は、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段とを有することを特徴とする通信装置を提供する。

【0009】

また、本発明は、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶手段と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得手段と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行手段と、前記記憶手段に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定手段と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び

前記識別情報に基づいて、前記特定手段に特定されたデータを内包するオブジェクトの型を選択する選択手段と、前記特定手段に特定されたデータを内包する、前記選択手段によって選択された型のオブジェクトを生成するオブジェクト生成手段と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行手段に許可する使用オブジェクト制御手段とを有することを特徴とする通信装置を提供する。

【0010】

また、本発明は、コンピュータに、データを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記特定機能に特定されたデータの使用について、前記オブジェクト生成機能により生成されたオブジェクトの使用のみを前記実行機能に許可するアクセス制御機能とを実現させるためのプログラムを提供する。

【0011】

また、本発明は、コンピュータに、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ

引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能とを実現させるためのプログラムを提供する。

【0012】

また、本発明は、コンピュータに、データと該データの保護に求められる確実さを示す重要度情報とを記憶する記憶機能と、該データを取り扱うメソッドが呼び出されるように記述されたプログラムと、該プログラムによって該データが適正に取り扱われる確率を示す識別情報とを取得する取得機能と、前記プログラムを実行するとともに、該プログラムについて使用が許可されたデータを使用する実行機能と、前記記憶機能に記憶されたデータのうち、前記プログラムについて使用が要求されるデータを特定する特定機能と、内包したデータを呼び出し元へ引き渡すメソッドを有するオブジェクトの型である非完全カプセル化型と該メソッドを有さないオブジェクトの型である完全カプセル化型との二つの型から、前記重要度情報及び前記識別情報に基づいて、前記特定機能に特定されたデータを内包するオブジェクトの型を選択する選択機能と、前記特定機能に特定されたデータを内包する、前記選択機能によって選択された型のオブジェクトを生成するオブジェクト生成機能と、前記識別情報で示される確率に応じた型のオブジェクトの使用を前記実行機能に許可する使用オブジェクト制御機能とを実現させるためのプログラムを提供する。

【0013】

この発明によれば、通信装置は、プログラムと該プログラムの識別情報を受信し、前記プログラムを実行した場合に使用されるデータを特定し、前記識別情報に基づいて生成するオブジェクトの型（非完全カプセル化型、または完全カプセル化型）を選択して、選択された型のオブジェクトを生成し、前記プログラムを

実行するときには、前記生成されたオブジェクトのみを使用して前記データを使用する。

【0 0 1 4】

【発明の実施の形態】

[1 . 第 1 実施形態]

以下、図面を参照して本発明の第 1 実施形態について説明する。なお、各図において共通する部分には、同一の符号が付されている。

【0 0 1 5】

[1 - 1 . 実施形態の構成]

< 1 - 1 - 1 . 通信システムの構成 >

図 1 は、この発明の実施形態に係る通信システム 1 の構成を示すブロック図である。同図に示すように通信システム 1 は、コンテンツサーバ 1 0 と、インターネット 2 0 と、移動パケット通信網 3 0 と、携帯電話機 4 0 とを有している。なお、この通信システム 1 には、本来、多数の携帯電話機 4 0 が収容されるが、図面が煩雑になることを防ぐため、図 1 には、1 つの携帯電話機 4 0 のみを図示している。また、同様の理由により、図 1 には、それぞれ 1 つのコンテンツサーバ 1 0、ゲートウェイサーバ 3 1 および基地局 3 2 のみを図示している。

【0 0 1 6】

コンテンツサーバ 1 0 は、インターネット 2 0 および移動パケット通信網 3 0 を介して携帯電話機 4 0 とパケット通信を行う機能を有している。このコンテンツサーバ 1 0 には、携帯電話機 4 0 に提供するプログラムや画像データ、楽曲データなどの種々のコンテンツが格納されている。これらのコンテンツの中には、携帯電話機 4 0 において実行可能な Java アプリケーションプログラム（以下、Java A P と略称する）がある。

【0 0 1 7】

移動パケット通信網 3 0 は、当該移動パケット通信網 3 0 に収容される携帯電話機 4 0 に対してパケット通信サービスを提供する通信網である。ゲートウェイサーバ 3 1 は、移動パケット通信網 3 0 とインターネット 2 0 とのデータの授受を中継する。また、基地局 3 2 は、移動パケット通信網 3 0 の通信サービスエリ

ア内に多数設置されており、携帯電話機 4 0 と無線通信を行う。

【 0 0 1 8 】

携帯電話機 4 0 は、基地局 3 2 と無線通信を行う。また、この携帯電話機 4 0 は、移動パケット通信網 3 0 およびインターネット 2 0 を介してコンテンツサーバ 1 0 とパケット通信を行う機能を有しており、コンテンツサーバ 1 0 からコンテンツをダウンロードすることができる。

【 0 0 1 9 】

< 1 - 1 - 2 . 携帯電話機の構成 >

図 2 は、携帯電話機 4 0 のハードウェア構成を示すブロック図である。同図に示すように携帯電話機 4 0 は、無線通信部 4 0 1 と、操作入力部 4 0 2 と、通話処理部 4 0 3 と、通信インタフェース 4 0 4 と、CPU 4 0 5 と、液晶表示部 4 0 6 と、記憶部 4 0 7 とを有しており、これらの各部はバス 4 1 1 により接続されている。

【 0 0 2 0 】

無線通信部 4 0 1 は、アンテナ 4 0 1 a を備え、基地局 3 2 との間で行われる無線通信を制御する。この無線通信部 4 0 1 は、CPU 4 0 5 の制御の下、送話音声に関するデータやパケット通信用のデータなどを搬送波に重畳して送信信号を生成し、この信号を基地局 3 2 へ送信する。また、無線通信部 4 0 1 は、基地局 3 2 から送られてくる無線信号をアンテナ 4 0 1 a を介して受信し、この信号を復調して自機 4 0 宛の受話音声に関するデータやパケット通信用のデータなどを得る。

【 0 0 2 1 】

操作入力部 4 0 2 は、数字や文字、操作指示などを入力するための複数のキーを有しており、これらのキーの操作に応じた操作信号を CPU 4 0 5 に出力する。また、通話処理部 4 0 3 は、例えば、マイクロフォンやスピーカ、音声処理部などを有しており、CPU 4 0 5 の制御の下、呼の接続／切断を含む通話処理を行う。

【 0 0 2 2 】

通信インタフェース 4 0 4 は、通信ケーブルを介して接続された電子機器との

有線通信を制御する。また、CPU 4 0 5 は、記憶部 4 0 7 に格納されている各種プログラムを実行することにより、バス 4 1 1 を介して接続されている装置各部を制御する。また、液晶表示部 4 0 6 は、液晶表示パネルと、この液晶表示パネルの表示制御を行う駆動回路とを有している。

【 0 0 2 3 】

記憶部 4 0 7 は、ROM 4 0 8 と、RAM 4 0 9 と、例えば、SRAM (Static-RAM) や EEPROM (Electrically Erasable Programmable-ROM) などの不揮発性メモリ 4 1 0 とを有している。ROM 4 0 8 には、例えば、携帯電話機 4 0 用のオペレーティングシステム (以下、OS と略称する) や Web (World Wide Web) ブラウザ等のソフトウェア、Java 実行環境を構築するためのソフトウェアが記憶されている。また、RAM 4 0 9 は、CPU 4 0 5 のワークエリアとして用いられ、CPU 4 0 5 により実行される各種のプログラムやデータが一時的に記憶される。

【 0 0 2 4 】

不揮発性メモリ 4 1 0 には、携帯電話機 4 0 の製品出荷時点から当該携帯電話機 4 0 用のプログラムが記憶される。また、不揮発性メモリ 4 1 0 には、コンテンツサーバ 1 0 からダウンロードされた Java A P などのコンテンツが記憶される。加えて、この不揮発性メモリ 4 1 0 には、電話番号を表すデータやメールアドレスを表すデータなどのデータを含んでいるアドレス帳データ、受信あるいは送信した電子メールデータ、着信や発信に関する履歴データ、電子決済を行うためのユーザの銀行口座番号を表すデータやクレジットカード番号を表すデータなどの各種データが記憶される。

【 0 0 2 5 】

なお、以下、本明細書では、携帯電話機 4 0 の製品出荷時点において既に ROM 4 0 8 や不揮発性メモリ 4 1 0 に記憶されているプログラムを、ダウンロードされた Java A P と区別するため、ネイティブプログラムと記載する。このネイティブプログラムには、自身がネイティブプログラムであることを示す識別情報が付与されている。

【 0 0 2 6 】

また、不揮発性メモリ 4 1 0 は、J A R ストレージ 4 1 0 a と、個別スクラッチパッド 4 1 0 b と、共通スクラッチパッド 4 1 0 c とを有している。

【0 0 2 7】

ここで、J A R ストレージ 4 1 0 a、個別スクラッチパッド 4 1 0 b および共通スクラッチパッド 4 1 0 c について説明する前に、まず、携帯電話機 4 0 にダウンロードされる Java A P について説明する。Java A P は、Java A P の本体プログラムおよび当該本体プログラムの実行に応じて利用される画像ファイルや音声ファイルなどを 1 つにまとめた J A R (Java Archive) ファイルと、この J A R ファイルのインストールや起動、ネットワークアクセスなどを制御するための各種制御情報が記述された A D F (Application Descriptor File) とを有している。ダウンロードされた J A R ファイルおよび A D F は、不揮発性メモリ 4 1 0 に記憶される。本実施形態においては、A D F には、図 3 に示すように、Java A P の名称を示す「AppName」や、インターネット 2 0 における J A R ファイルの U R L を示す「PackageURL」や、J A R ファイルのサイズをしめす「AppSize」や、J A R ファイルの最終更新日を示す「LastModified」等の従来から A D F に内包されているデータに加えて、「トラステッドアプリケーション識別子」が内包されている。この「トラステッドアプリケーション識別子」とは、移動パケット通信網 3 0 を運営する通信事業者や C A (Certificate Authority) のような公正な第 3 者機関により Java A P の内容が審査され、一定の基準を満たしていると認定された Java A P とそれ以外のプログラムとを識別するためのデータである。一定の基準とは、例えば、プログラムが携帯電話機 4 0 に記憶されているデータを外部に漏洩させることなく適正に取り扱い、携帯電話機 4 0 において正常に動作する、等の基準である。上記の第 3 者機関は、移動パケット通信網 3 0 を用いて提供される通信サービスに加入した全ての者に信頼されているから、この第 3 者機関により認定されたプログラムには、一定の信頼が化体する。よって、ここでは、このような一定の信頼が化体したプログラムを「トラステッドアプリケーション」とよび、それ以外のプログラムを「非トラステッドアプリケーション」とよぶ。「トラステッドアプリケーション識別子」は、その値が"0"である場合には、当該 A D F ファイルに対応する Java A P が非トラステッドアプリケーション

ョンであることを示し、“1”である場合には、当該 A D F ファイルに対応する Java A P がトラステッドアプリケーションであることを示している。

【 0 0 2 8 】

J A R ストレージ 4 1 0 a および個別スクラッチパッド 4 1 0 b には、ダウンロードされた Java A P 毎に当該 Java A P 用の記憶領域が設けられる。J A R ストレージ 4 1 0 a 内の各記憶領域には、Java A P の J A R ファイルが記憶される。また、個別スクラッチパッド 4 1 0 b 内の各記憶領域には、例えば、Java A P がゲームプログラムである場合、過去の得点データやセーブデータなど、Java A P の利用に応じて発生した当該 Java A P 用のデータが記憶される。さらに、共通スクラッチパッド 4 1 0 c には、複数の Java A P が共通して使用するデータが記憶される。

【 0 0 2 9 】

また、ダウンロードの後、Java A P が携帯電話機 4 0 において実行される場合、この Java A P の実行に伴って携帯電話機 4 0 がアクセスすることのできるリソースは、この Java A P のダウンロード元のコンテンツサーバ 1 0 と、この Java A P に対して割り当てられた J A R ストレージ 4 1 0 a および個別スクラッチパッド 4 1 0 b 内の記憶領域と、共通スクラッチパッド 4 1 0 c と、のみに制限され、それ以外のリソースにアクセスすることはできない。

【 0 0 3 0 】

< 1 - 1 - 3 . Java 実行環境 >

図 4 は、携帯電話機 4 0 における Java A P の実行環境を説明するための図である。同図に示すように本実施形態に係る携帯電話機 4 0 には、Java A P の実行環境を構築するためのソフトウェアとして、K V M (K Virtual Machine) と、コンフィギュレーションとして C L D C (Connected Limited Device Configuration) と、プロファイルとして通信事業者が独自に策定したオリジナル拡張ライブラリとが記憶されている。

【 0 0 3 1 】

K V M は、小型電子機器用に設計変更された J V M (Java Virtual Machine) であって、Java A P の実行ファイル形式であるバイトコードを C P U 4 0 5 が O

Sを介して解釈／実行可能な命令コードに変換する。また、CLDCクラスライブラリは、CLDC用のクラスライブラリである。

【0032】

オリジナル拡張ライブラリは、CLDCを基礎として携帯電話機に特化した機能を提供するためのクラスライブラリである。このオリジナルJava拡張ライブラリには、例えば、ユーザインタフェースAPI (Application Program Interface)、ネットワーキングAPI、スクラッチパッドAPI、完全カプセル化API、非完全カプセル化APIなどが含まれている。

【0033】

ここで、ユーザインタフェースAPIは、携帯電話機40のユーザインタフェース機能をサポートするAPIであり、ネットワーキングAPIは、URL (Uniform Resource Locator) により指定されたネットワークリソースへのアクセスをサポートするAPIである。また、スクラッチパッドAPIは、個別スクラッチパッド410cや共通スクラッチパッド410dに対するデータの書き込みや読み出しをサポートするAPIである。さらに、完全カプセル化APIは、完全カプセル化オブジェクトを生成するためのAPIであり、非完全カプセル化APIは、非完全カプセル化オブジェクトを生成するためのAPIである。

【0034】

また、携帯電話機40は、CLDCクラスライブラリおよびオリジナル拡張ライブラリに加え、メーカー独自拡張ライブラリを有している。このメーカー独自拡張ライブラリは、携帯電話機40を製造する各メーカーがそれぞれ独自の機能を提供するためのクラスライブラリである。

【0035】

次に、JAM (Java Application Manager) は、OSによる制御の下で、携帯電話機40にダウンロードされたJavaAPや、完全カプセル化オブジェクト、非完全カプセル化オブジェクトなどを管理する機能を有している。例えば、JAMは、JavaAPのインストールや更新、削除を行う機能、不揮発性メモリ410に記憶されているJavaAPをリスト表示する機能、JavaAPの実行管理（起動や強制終了など）を行う機能、JavaAPの実行に伴う携帯電話機40のアクセスを制

限する機能、完全カプセル化オブジェクトや非完全カプセル化オブジェクトの生成、更新、削除を行なう機能などを有している。

【0036】

また、同図に示すように、電話帳機能やブラウザ機能、ネットワーク通信機能などを提供するネイティブプログラムは、OSによる制御の下で直接動作する。

【0037】

<1-1-4. オブジェクトの構成>

次に、オブジェクトについて説明する。オブジェクトとは、データ（Javaプログラミング言語においてはフィールド）と操作（Javaプログラミング言語においてはメソッド）が一体となったものである。Javaプログラミング言語では、「private」というアクセス修飾子を用いてオブジェクト内のフィールドをprivateフィールドに宣言することで、当該privateフィールドに記憶されるデータのカプセル化を図る。このカプセル化により、カプセル化オブジェクトが生成される。

図5は、カプセル化オブジェクトについて説明するための模式図である。同図に示すように、カプセル化オブジェクトとは、カプセル化された1以上のデータと、当該カプセル化された各データに対するオブジェクト外部からの操作を可能とするための1以上のメソッドとを有するオブジェクトである。

【0038】

同図に示す例では、2つのデータ1，2と、2つのメソッド1，2とを有するカプセル化オブジェクトが例示されている。このカプセル化オブジェクト内のデータ1，2は共にカプセル化されているため、オブジェクトの外部からデータ1，2を直接読み書きすることはできない。したがって、ダウンロードされたプログラムがカプセル化オブジェクト内のデータ1，2に対してアクセスする場合、プログラムは、メソッド1，2を使用して目的のデータ1またはデータ2に対する操作をカプセル化オブジェクトに指令しなければならない。

【0039】

ここで、同図に示すメソッド1が、例えば、指定されたデータ自体を操作元のプログラムへ引き渡すメソッドであれば、操作元のプログラムは、メソッド1を使用してカプセル化オブジェクト内の任意のデータ1，2を取得することが可能

である。また、同図に示すメソッド2が、例えば、指定されたデータを液晶画面に表示させるメソッドであれば、操作元のプログラムは、メソッド2を使用してカプセル化オブジェクト内の任意のデータ1, 2を画面表示させることが可能である。ここで注目すべき点は、メソッド2を使用してカプセル化オブジェクト内の任意のデータ1, 2を画面表示させたプログラムは、表示させるデータやメソッドを指定してカプセル化オブジェクトへ指令を出すものの、表示させるデータ自体を取得していない点である。

【0040】

つまり、データそのものを操作元のプログラムに引き渡すメソッドを1つも有していないカプセル化オブジェクト（完全カプセル化オブジェクト）であれば、操作元のプログラムは、オブジェクト内のデータそのものを取得することはできないが、このオブジェクトに備わるメソッドを使用してオブジェクト内のデータ操作を行うことはできる。

【0041】

したがって、操作元のプログラムが、非トラステッドアプリケーションである場合には、プログラムがアクセスするデータ全てを完全カプセル化オブジェクトとして扱うようにすれば、当該プログラムにデータ自体を引き渡すことがないので、携帯電話機40に記憶されているデータのセキュリティを確保することができる。また、このように扱うことで、非トラステッドアプリケーションであったとしても、アドレス帳データや電子メールデータなど、従来はセキュリティを確保する観点から一切のアクセスを認めていなかったデータに対し、完全カプセル化オブジェクトが有するメソッドを用いて操作（アクセス）を行うことができる。

【0042】

図6は、電話帳データに関する非完全カプセル化オブジェクトについて例示する模式図である。

上述したように、Javaプログラミング言語では、「private」というアクセス修飾子を用いてオブジェクト内のフィールドをprivateフィールドに宣言することで、当該privateフィールドに記憶されるデータのカプセル化を図る。つまり

、オブジェクト内のフィールドが全てprivateフィールドである場合、各privateフィールドに記憶されているデータをオブジェクトの外部から直接読み書きすることができなくなる。このようにした場合、各privateフィールドに記憶されているデータに対してオブジェクトの外部からアクセスするには、このオブジェクトに備わるメソッドを使用してデータに対する操作を当該オブジェクトに指令しなければならない。

【 0 0 4 3 】

同図に示す非完全カプセル化オブジェクトには、2つのprivateフィールドが設けられ、それぞれprivate char value[1]、private char value[2]という電話帳の文字列データが記憶されている。また、この非完全カプセル化オブジェクトは、getBytes()、drawString()という2つのメソッドを有している。ここで、getBytes()は、オブジェクト内のデータをバイト配列の形式で操作元のプログラムへ引き渡すメソッドである。したがって、ダウンロードされたJavaA Pは、このgetBytes()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ (private char value[1]、private char value[2]) を取得することが可能である。加えて、JavaA Pは、取得した電話帳の文字列データを当該JavaA Pのダウンロード元のサーバ (コンテンツサーバ10) へ送信することなどができる。

【 0 0 4 4 】

また、drawString()は、オブジェクト内のデータを携帯電話機40の液晶画面に表示させるメソッドである。JavaA Pは、このdrawString()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ (private char value[1]、private char value[2]) を液晶画面に表示させることもできる。

【 0 0 4 5 】

一方、図7は、電話帳データに関する完全カプセル化オブジェクトについて例示する模式図である。同図に示す完全カプセル化オブジェクトが図6に示した非完全カプセル化オブジェクトと異なるのは、完全カプセル化オブジェクトは、上述したgetBytes()のように、オブジェクト内のデータそのものを操作元のプログ



ラムへ引き渡すメソッドを有していない点である。

【0046】

すなわち、完全カプセル化オブジェクトは、カプセル化された上に、オブジェクト内のデータそのものを操作元のプログラムへ引き渡すメソッドを1つも有していない。したがって、ダウンロードされたJavaA Pは、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データ (private char value[1]、private char value[2]) を画面表示させることはできるが、電話帳の文字列データそのものを取得することはできない。以上のようなことから、非トラステッドアプリケーションが携帯電話機40にダウンロードされた場合であっても、このような非トラステッドアプリケーションに電話帳データが引き渡されることはなく、当然、電話帳データがサーバなど携帯電話機40の外部へ送信されることもない。

【0047】

ところで、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データを画面表示させる場合、完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、ネイティブプログラムとしてROM408または不揮発性メモリ410に記憶されている表示制御プログラムを使用して液晶画面に電話帳の文字列を表示させる。この表示制御プログラムからJavaA Pが表示データを取得することができてしまうと、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いた意味がなくなってしまう。

【0048】

しかしながら、ダウンロードされたJavaA Pが実行される場合には、前述したようにJAMのアクセス制限機能により、JavaA Pの実行中に携帯電話機40がアクセスすることのできるリソースが制限される。ここで、JavaA Pの実行中に携帯電話機40のアクセスが許可されるリソースに表示制御プログラムは含まれていないので、JavaA Pが表示制御プログラムから表示データを取得するようなことは一切あり得ない。

【0049】

また、カプセル化は、プログラミング言語レベルでのカプセル化と、実行コー

ド（マシン語またはバイトコード）レベルでのカプセル化とが考えられる。プログラミング言語レベルでのカプセル化が完全であっても、実行コードレベルでのカプセル化が完全でなければ、データを完全にカプセル化したとは言えない。例えば、プログラミング言語であるC++を用いたプログラムでも privateフィールドを有するカプセル化オブジェクトを生成することはできる。しかしながら、C++は、プログラミング言語レベルでの完全カプセル化しか達成し得ない。

【0050】

具体的に説明すると、C++を用いたプログラムにより、オブジェクト内の全てのフィールドを privateフィールドとして宣言し、カプセル化オブジェクトを生成した場合、このオブジェクト内の privateフィールドに記憶されているデータを直接読み書きするようなソースコードをコンパイルしようとしてもコンパイルエラーとなり、実行コードは生成されない。

しかしながら、これは、コンパイラによって保証されているに過ぎない。例えば、悪意のある第3者がコンパイラを改造することで、オブジェクト内の privateフィールドに記憶されているデータを直接読み書きするような実行コードを生成することも可能である。また、コンパイラを改造しなくても、悪意のある第3者がハンドアセンブルなどの手段でオブジェクト内のデータを不正に読み出す実行コードを生成するプログラムを作成することも不可能ではない。加えて、ポインタを用いて直接メモリにアクセスしてしまえば、オブジェクト内のデータを入手することができてしまう。

【0051】

これに対してJavaの場合、private宣言されたフィールドは、private属性を有するフィールドであることを示すJavaのバイトコードへコンパイルされる。KVMがクラスファイルをRAM 409などへ展開する際も、フィールドの private属性は保持されている。したがって、仮にコンパイラを改造してオブジェクト内の privateフィールドに記憶されているデータを不正に読み出すようなバイトコードを生成したとしても、KVMまたはJAMがこれを検知するので、オブジェクト内のデータを入手することはできない。また、Javaはポインタをサポートしていないので、ポインタを用いて直接メモリにアクセスし、オブジェクト内の

データを手に入れることもできない。

【0052】

以上のようなことから、Javaでは、プログラミング言語レベルのみに止まらず、バイトコードレベルでの完全なカプセル化を達成することが可能である。

【0053】

[1-2. 実施形態の動作]

次に、本実施形態の動作について説明する。

なお、携帯電話機40が以下に述べる動作を行う前提として、携帯電話機40は、移動パケット通信網30およびインターネット20を介してコンテンツサーバ10とパケット通信を行い、コンテンツサーバ10からJavaAPをダウンロードして不揮発性メモリ410に記憶しているものとする。また、不揮発性メモリ410には、ダウンロードされたJavaAP（JARファイルとADFファイル）の他に、アドレス帳データや電子メールデータ、ユーザデータなどが記憶されているものとする。

【0054】

<1-2-1. オブジェクト生成処理>

まず、携帯電話機40においてCPU405により実行されるオブジェクト生成処理について図8を参照して説明する。このオブジェクト生成処理は、JAMの機能としてCPU405により実行されるものであり、例えば、画面表示されたプログラムの一覧リストの中から、実行するプログラムが操作入力により指定された場合などに実行される。なお、プログラムの実行を指示する形態は、操作入力によるものに限定されず、例えば、予め定められた時間毎にプログラムの実行が指示される場合や、既に実行されている他のプログラムから実行が指示される場合、電子メールなどを用いて携帯電話機40の外部からプログラムの実行が指示される場合などもある。

【0055】

同図に示すように、まず、携帯電話機40のCPU405は、実行するプログラムとして操作入力により指定されたプログラムを特定する（ステップS101）。次いで、CPU405は、特定したプログラムがダウンロードされたJavaA

Pであるのか、それともネイティブプログラムであるのかを判別する（ステップ S 1 0 2）。前述したようにネイティブプログラムには、自身がネイティブプログラムであることを示す識別情報が付与されている。したがって、CPU 4 0 5 は、プログラムに上記識別情報が付与されているか否かを判別することで、このプログラムがダウンロードされたJava A Pであるのか、それともネイティブプログラムであるのかを判別することができる。

【0 0 5 6】

その結果、CPU 4 0 5 は、プログラムがネイティブプログラムであると判別した場合には（ステップ S 1 0 2：N o）、オブジェクト生成処理を終了するとともに、実行するプログラムとして指定されたネイティブプログラムを起動する。そして、CPU 4 0 5 は、起動させたネイティブプログラムに基づく処理を行なう。

【0 0 5 7】

ここで、実行するプログラムがネイティブプログラムである場合は、携帯電話機に記憶されたデータを適法かつ適正に取り扱うことが携帯電話機の製造者により保証されているから、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いたり、あるいはネイティブプログラムの実行に伴ってJ A Mのアクセス制限機能を動作させる必要がない。したがって、ネイティブプログラムが実行される場合、J A Mによるアクセス制限は一切行われず、ネイティブプログラムは、携帯電話機 4 0 内の任意のリソースおよびネットワーク上の任意のリソースにアクセスすることができる。

【0 0 5 8】

一方、CPU 4 0 5 は、プログラムがダウンロードされたJava A Pであると判別した場合は（ステップ S 1 0 2：Y e s）、次いで、不揮発性メモリ 4 1 0 に記憶されている各種のデータの中から、このJava A Pを実行した場合に使用されるデータを、例えば、このJava A Pのプログラム内容を解析するなどして特定する（ステップ S 1 0 3）。なお、Java A Pが使用するデータを特定する際には、J A Rストレージ 4 1 0 a 内の、このJava A PのJ A R ファイルに記憶されているデータは特定の対象から除外する。これは、J A R ファイル内に記憶されてい

るデータは、このJava A Pを実行する上で必要となるデータとして当該Java A Pを提供するコンテンツプロバイダが用意したデータであるためである。

【0059】

次いで、CPU 405は、ADFファイルに内包されているトラステッドアプリケーション識別子を参照して、上記特定したデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを決定する（ステップS104）。例えば、CPU 405は、トラステッドアプリケーション識別子が”1”である場合には、当該ADFファイルに対応するJava A Pはトラステッドアプリケーションであるため、上記特定したデータを扱うオブジェクトの型を「完全カプセル化」型に決定する。

【0060】

この後、CPU 405は、上記ステップS103において特定したデータと、上記ステップS104において決定したオブジェクトの型とに基づいて、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成する（ステップS105）。例えば、上記ステップS104においてデータを扱うオブジェクトの型を「完全カプセル化」型に決定した場合は、CPU 405は、オリジナル拡張ライブラリ内の完全カプセル化APIを起動して、特定されたデータ全てについて完全カプセル化オブジェクトを生成する。また、上記ステップS104においてデータを扱うオブジェクトの型を「非完全カプセル化」型に決定した場合は、CPU 405は、オリジナル拡張ライブラリ内の非完全カプセル化APIを起動して、特定されたデータ全てについて非完全カプセル化オブジェクトを生成する。

【0061】

次いで、CPU 405は、生成した完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを共通スラッチパッド410dに記憶し（ステップS106）、オブジェクト生成処理を終了する。なお、上記ステップS105において生成された完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、共通スラッチパッド410dではなく、個別スラッチパッド410cに記憶される形態であってもよい。

【 0 0 6 2 】**< 1 - 2 - 2 . アクセス管理処理 >**

次に、携帯電話機 4 0 において CPU 4 0 5 により実行されるアクセス管理処理について図 9 を参照して説明する。このアクセス管理処理は、J A M の機能として CPU 4 0 5 により実行されるものであり、ダウンロードされた Java A P の実行過程においてアクセス要求が発生した場合に、割り込み処理として実行される。

【 0 0 6 3 】

同図に示すように、まず、携帯電話機 4 0 の CPU 4 0 5 は、Java A P の実行過程において発生したアクセス要求について、要求されたアクセス先が予め許可された範囲内のリソースであるか否かを判別し、アクセスを許可するか否かを判定する（ステップ S 2 0 1）。ここで、アクセスの許可／不許可を判定する仕組みについて具体的に説明すると、ダウンロードされた Java A P が実行される場合、CPU 4 0 5 は、Java A P の実行に伴ってアクセスすることのできるリソースを、この Java A P の A D F に記述されている URL により指定される当該 Java A P のダウンロード元のコンテンツサーバ 1 0 と、この Java A P に対して割り当てられた J A R ストレージ 4 1 0 b および個別スクラッチパッド 4 1 0 c 内の記憶領域と、共通スクラッチパッド 4 1 0 d と、のみに制限する。

【 0 0 6 4 】

したがって、CPU 4 0 5 は、要求されたアクセス先が上述したリソースのいずれかである場合は、このアクセスを許可する一方、要求されたアクセス先が上述したリソース以外である場合は、このアクセス要求を許可しない。

【 0 0 6 5 】

次いで、CPU 4 0 5 は、アクセスの許可／不許可を示す判定結果を要求元の Java A P に通知した後（ステップ S 2 0 2）、アクセス管理処理を終了する。また、実行中の Java A P は、J A M による判定結果を受け取ると、この判定結果に従って、アクセスが許可された場合は当該アクセス要求に基づく処理を実行する一方、アクセスが許可されなかった場合は当該アクセス要求に基づく処理をキャンセルする。

【 0 0 6 6 】

さて、携帯電話機 4 0 の CPU 4 0 5 は、ダウンロードした Java A P を実行する場合、図 8 に示したオブジェクト生成処理を行った後に Java A P を起動する。また、ダウンロードした Java A P の実行過程において CPU 4 0 5 は、アクセス要求が発生すると、図 9 に示したアクセス管理処理を行う。したがって、携帯電話機 4 0 は、ダウンロードした Java A P の実行中において必ず J A M によるアクセス制限を受けることとなり、例えば、不揮発性メモリ 4 1 0 に記憶されているアドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータ、コンテンツなどのデータそのものにアクセスすることができなくなる。

【 0 0 6 7 】

このため、携帯電話機 4 0 の CPU 4 0 5 は、上述したオブジェクト生成処理において、起動させる Java A P が使用するデータを特定し、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド 4 1 0 d に記憶する。この共通スクラッチパッド 4 1 0 d は、前述したように、J A M によるアクセス制限が行われている場合であっても携帯電話機 4 0 のアクセスが許可されるリソースである。また、携帯電話機 4 0 にダウンロードされる Java A P は、共通スクラッチパッド 4 1 0 d に記憶された完全カプセル化オブジェクトや非完全カプセル化オブジェクトにアクセスし、当該オブジェクトに備わるメソッドを使用してこのオブジェクト内のデータに対する操作を指令するように作成されている。

【 0 0 6 8 】

例えば、アドレス帳データを使用する非トラステッドアプリケーションが起動される場合、上述したオブジェクト生成処理によりアドレス帳データ用の完全カプセル化オブジェクトが生成され、共通スクラッチパッド 4 1 0 d に記憶される。また、この非トラステッドアプリケーションは、上記生成されたアドレス帳データ用の完全カプセル化オブジェクトに対して、当該オブジェクトに備わるメソッドを用いてこのオブジェクト内のデータに対する操作を指令する。したがって、完全カプセル化オブジェクトの有するアドレス帳データの一部を画面表示させることなどが可能となる一方、完全カプセル化オブジェクトの有するデータその

ものが非トラステッドアプリケーションに引き渡されることはない。

【0069】

従来は、ダウンロードされたJavaA Pに対するセキュリティを確保するため、このようなJavaA Pについては、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどに一切アクセスすることができなかった。これに対して本実施形態によれば、完全カプセル化オブジェクトを用いることにより、データそのものがJavaA Pに引き渡されることがないので、ダウンロードされたJavaA Pに対するセキュリティを確保しつつ、同時に、従来は一切アクセスできなかったデータについて、完全カプセル化オブジェクトを介して画面表示を行わせることなどができるようになる。したがって、ダウンロードされたJavaA Pが携帯電話機40において実現することのできる機能を充実させることができる。

また、本実施形態によれば、JavaAPを作成するプログラマは、オブジェクトを使用してデータにアクセスするプログラムをコーディングすることができるために、データへのアクセス方法やセキュリティを考慮せずにプログラムのコーディングを行うことができる。これにより、プログラマの開発生産性や保守性が向上する。

【0070】

<1-2-3. JavaA P 終了処理>

次に、携帯電話機40においてCPU405により実行されるJavaA P 終了処理について図10を参照して説明する。このJavaA P 終了処理は、JAMの機能としてCPU405により実行されるものであり、JavaA Pの実行終了要求が発生した場合に、割込処理として実行される。

【0071】

同図に示すように、携帯電話機40のCPU405は、JavaA Pの実行終了要求が発生すると、共通スクラッチパッド410dに記憶されている完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除する（ステップS301）。このステップS301において削除される完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、JavaA Pを起動させる際に、上述したオブジェ

クト生成処理（図 8 参照）において生成され、共通スクラッチパッド 4 1 0 d に記憶されたものである。CPU 4 0 5 は、共通スクラッチパッド 4 1 0 d からオブジェクトを削除すると、JavaA P 終了処理を終える。

【0 0 7 2】

このようにダウンロードされたJavaA P を起動する際に、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを生成して共通スクラッチパッド 4 1 0 d に記憶する一方、このJavaA P の実行が終了する際に、共通スクラッチパッド 4 1 0 d から完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除するようにすれば、携帯電話機 4 0 のメモリ資源を効率的に活用することができる。

【0 0 7 3】

[2 . 第 2 実施形態]

上記第 1 実施形態においては、JavaA P がトラステッドアプリケーションか否かによって、データの種類に関わらず一律に完全カプセル化オブジェクトを生成するか、非完全カプセル化オブジェクトを生成するかを決定したが、この第 2 実施形態においては、JavaA P に化体している信頼のレベルとデータの保護に求められる確実さを示す重要度とによって、完全カプセル化オブジェクトを生成するか、非完全カプセル化オブジェクトを生成するかを決定する。さらに、JavaA P の信頼のレベルによって、使用可能なオブジェクトを決定する。

【0 0 7 4】

[2 - 1 . 実施形態の構成]

第 1 実施形態における、ADF の「トラステッドアプリケーション識別子」の値は、一定の信頼が化体していないJavaA P の場合には“ 0 ”、一定の信頼が化体しているJavaA P の場合には“ 1 ”が設定されていたが、第 2 実施形態においては、JavaA P に化体している信頼のレベル（以下、「JavaA P の信頼度」という）に応じて、“高”、“中”、“低”のレベルが設定される。ここで、例えば、JavaA P に化体している信頼度が高いとは、JavaA P によってデータが適正に取り扱われる確率が予め定められた基準確率より高いことを意味する。

【0 0 7 5】

＜ 2 - 1 - 1 . 携帯電話機の構成 ＞

携帯電話機 4 0 の不揮発性メモリ 4 1 0 には、図 1 1 に示すように、重要度テーブル 4 1 0 d が設けられている。

同図に示すように、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータのような、携帯電話機 4 0 に記憶されているデータの中でも特にデータを確実に保護する必要が求められているデータには、重要度の値が“高”に設定される。また、求められる保護の確実さが中間のデータには重要度の値が“中”、低いデータには重要度の値が“低”に設定される。

また、携帯電話機 4 0 の不揮発性メモリ 4 1 0 には、図 1 2 に示すように、アプリデータ関係テーブル 4 1 0 e が設けられている。このテーブルには、データの重要度と JavaA P の信頼度との組み合わせによって、当該データを完全カプセル化型として扱うのか、それとも非完全カプセル化型として扱うのかが設定されている。例えば、同図において、信頼度が高い JavaA P の場合は、データの重要度に関わらず非カプセル化型が設定されている。また、信頼度が低い JavaA P の場合は、データの重要度が“高”および“中”の場合に完全カプセル化型が設定され、データの重要度が“低”の場合に非完全カプセル化型が設定されている。

また、このアプリデータ関係テーブル 4 1 0 e には、JavaA P の信頼度によって、使用可能なオブジェクトが設定されている。例えば、同図において、信頼度が高い JavaA P は、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトを使用可能である。よって、JavaA P は、JavaA P 実行時に、データの重要度に関わらず生成される非完全カプセル化オブジェクトを使用することができる。また、信頼度が低い JavaA P も、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトを使用可能である。よって、JavaA P は、JavaA P 実行時に、データの重要度が“高”または“中”の場合に生成される完全カプセル化オブジェクトとデータの重要度が“低”の場合に生成される非完全カプセル化オブジェクトとを使用することができる。

なお、以上説明した重要度テーブル 4 1 0 d およびアプリデータ関係テーブル 4 1 0 e の内容は、携帯電話機 4 0 出荷時に予め登録されているが、サーバからダウンロードされるコンテンツは、ダウンロード時に重要度テーブル 4 1 0 d に

データが記憶される。また、ユーザが携帯電話機 4 0 を操作して、これらのテーブルに値を入力することが可能である。

以上説明した構成以外の構成は、第 1 実施形態と同様であるため、重複した説明を省略する。

【 0 0 7 6 】

[2 - 2 . 実施形態の動作]

次に、本実施形態の動作について説明する。

不揮発性メモリ 4 1 0 には、ダウンロードされた JavaA P の他に、アドレス帳データや電子メールデータ、ユーザデータなどが記憶されており、重要度テーブル 4 1 0 d およびアプリデータ関係テーブル 4 1 0 e には図 1 1 および図 1 2 の情報内容が記憶されているものとする。

【 0 0 7 7 】

< 2 - 2 - 1 . オブジェクト生成処理 >

次に、オブジェクト生成処理を、第 1 実施形態における図 8 に示すフローチャートを用いて説明する。

ステップ S 1 0 1 ~ ステップ S 1 0 3 までは、第 1 実施形態と同様である。

次に、C P U 4 0 5 は、不揮発性メモリ 4 1 0 より JavaA P に対応する A D F のトラステッド識別子を参照して、JavaA P の信頼度を取得する。次いで、C P U 4 0 5 は、重要度テーブル 4 1 0 d を参照して、ステップ S 1 0 3 で特定したデータの重要度を取得する。次いで、C P U 4 0 5 は、アプリデータ関係テーブル 4 1 0 e を参照して、データの重要度と JavaA P の信頼度とに基づいて、このデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを決定する（ステップ S 1 0 4）。例えば、JavaA P の使用するデータがアドレス帳データの場合、C P U 4 0 5 は、重要度テーブル 4 1 0 d から、アドレス帳データの重要度“高”を読み出す。また、例えば、JavaA P に対応する A D F のトラステッド識別子から取得した信頼度が“低”である場合、アプリデータ関係テーブル 4 1 0 e より、このアドレス帳データを扱うオブジェクトの型を「完全カプセル化」型に決定する。

【 0 0 7 8 】

この後、CPU405は、上記ステップS103において特定したデータと、上記ステップS104において決定したオブジェクトの型とに基づいて、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成する（ステップS105）。上記ステップS103において決定したオブジェクト型が完全カプセル化型の場合、CPU405は、オリジナル拡張ライブラリ内の完全カプセル化APIを起動して、完全カプセル化オブジェクトを生成する。また、上記ステップS103において決定したオブジェクト型が非完全カプセル化型の場合、CPU405は、オリジナル拡張ライブラリ内の非完全カプセル化APIを起動して、非完全カプセル化オブジェクトを生成する。

次いで、CPU405は、生成した完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを共通スクラッチパッド410dに記憶し（ステップS106）、オブジェクト生成処理を終了する。

【0079】

また、上記ステップS103においてJavaAPの使用データが複数特定された場合は、特定した各データ毎に、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド410dに記憶するため、上記ステップS104～S106までの処理を各データ毎に繰り返して行う。そして、CPU405は、オブジェクト生成処理を終了した後、実行するプログラムとして指定されたJavaAPを起動し、このプログラムに基づく処理を開始する。

【0080】

<2-2-2. オブジェクト使用管理処理>

次に、携帯電話機40においてCPU405により実行されるオブジェクト使用管理処理について図13を参照して説明する。

【0081】

同図に示すように、まず、携帯電話機40のCPU405は、JavaAPの実行過程においてオブジェクトを使用するための要求が発生した際、当該オブジェクトがJavaAPに使用が許可されているか否かをアプリデータ関係テーブル410eを参照することにより判別し、オブジェクトの使用を許可するか否かを判定す

る（ステップ S 4 0 1）。ここでは、図 1 2 のアプリデータ関係テーブル 4 1 0 e に示す通り、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトの双方とも使用可能であるため、オブジェクトの使用は許可される。

【 0 0 8 2 】

次いで、CPU 4 0 5 は、オブジェクト使用の許可／不許可を示す判定結果を要求元の Java A P に通知した後（ステップ S 4 0 2）、オブジェクト使用管理処理を終了する。また、実行中の Java A P は、判定結果を受け取ると、この判定結果に従って、使用が許可された場合は当該使用するための要求に基づく処理を実行する一方、使用が許可されなかった場合は当該使用するための要求に基づく処理をキャンセルする。

ここでは、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトとも使用が許可され、Java A P は、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを使用して、データにアクセスする。

【 0 0 8 3 】

上述した動作以外の動作（アクセス管理処理、Java A P 終了処理）は、第 1 実施形態と同様であるため、重複した説明を省略する。

【 0 0 8 4 】

以上のように、Java A P の信頼度とデータの重要度とによって、生成するオブジェクトを決定し、さらに、Java A P の信頼度によって使用可能なオブジェクトを決定しているため、データと Java A P との様々な組合わせに応じて、Java A P がデータに対してアクセスすることのできる許可レベル（データ自体を Java A P に引き渡す、引き渡さず取得するのみ、取得もしない）を設定することができる。このため、データに対するセキュリティを保持しながら、様々な Java A P を携帯電話機 4 0 で動作させることが可能となる。また、この許可レベルの設定は、重要度テーブル 4 1 0 d およびアプリデータ関係テーブル 4 1 0 e を使用して行うことができる。そして、Java A P を作成するプログラマは、データにアクセスするプログラムをコーディングする際に、セキュリティやデータへのアクセス方法を考慮せずに、これらのテーブルによって許可レベルが予め設定されているオブジェクトを使用すればよい。このため、プログラマの開発生産性や保守性が向

上する。

【0085】

[3. 変形例]

以上、本発明の実施形態について説明したが、本発明はその主要な特徴から逸脱することなく他の様々な形態で実施することが可能である。上述した実施形態は、本発明の一態様を例示したものに過ぎず、本発明の範囲は、特許請求の範囲に示す通りであって、また、特許請求の範囲の均等範囲に属する変形や変更は、全て本発明の範囲内に含まれる。なお、変形例としては、例えば、以下のようなものが考えられる。

【0086】

(1) 上記第2実施形態において使用したアプリデータ関係テーブル410eに設定した内容は、一例に過ぎず、例えば、図14に示すような内容をアプリデータ関係テーブル410eに設定することもできる。

この場合、CPU405は、このアプリデータ関係テーブル410eを参照することによって、トラステッドアプリケーション識別子の信頼度に関わらずデータの重要度が“高”の場合に、完全カプセル化オブジェクトを生成し、データの重要度が“中”および“低”の場合に、非完全カプセル化オブジェクトを生成する。そして、JavaAPは、JavaAP実行の際に、JavaAPに対応するトラステッドアプリケーション識別子の信頼度が“低”である場合には、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクト共に使用せず、トラステッドアプリケーション識別子の信頼度が“中”である場合には、完全カプセル化オブジェクトのみを使用し、トラステッドアプリケーション識別子の信頼度が“高”である場合には、完全カプセル化オブジェクトおよび非完全カプセル化オブジェクトの双方を使用する。このようにすることで、CPU405は、生成するオブジェクトをデータの重要度のみで決定することができ、生成したオブジェクトを使用する際には、JavaAPに化体した信頼のレベルのみによって、生成したオブジェクトを使用するか否かを決定することができる。

【0087】

(2) 上述した第2実施形態では、重要度テーブル410dとアプリデータ関係

テーブル 410e を用いる構成としたが、これらのテーブルは、データ構成の概念を示すものに過ぎず、通信装置に応じて最適なデータ構成を選択することができる。例えば、アドレス帳データ、電子メールデータ、コンテンツなどのデータ自体に重要度を示すデータを付与するようにすれば、重要度テーブル 410d を用いる必要はない。

【0088】

(3) 上記実施形態においては、ダウンロードされた JavaA P が、トラステッドアプリケーションか否かを識別したり、JavaA P の信頼度を識別するのに、JavaA P に対応する ADF に内包されているトラステッドアプリケーション識別子を用いたが、これは一例に過ぎない。例えば、JavaA P の信頼度や、JavaA P がトラステッドアプリケーションとして認定済である、等の JavaA P に化体した信頼に関するデータを管理するための管理サーバ装置を設置して、携帯電話機 40 の CPU 405 が、実行しようとする JavaA P に化体した信頼に関するデータが管理サーバに存在している場合には、当該データを管理サーバより受信するようにしてもよい。

また、上記実施形態においては、インターネット 20 に接続されたコンテンツサーバ 10 よりダウンロードされた JavaA P を用いた場合について説明したが、本発明は、ネイティブプログラム以外の任意のプログラム、すなわち携帯電話機 40 の販売後にそのメモリに記憶されるプログラムに用いた場合にも効果を奏する。例えば、赤外線インターフェースを備えた携帯電話機 40 が、赤外線インターフェースを備えたパーソナルコンピュータ等の通信機器より赤外線通信によってプログラムを受信し、そのプログラムに化体している信頼に関するデータを管理サーバから受信するようにしてもよい。

【0089】

(4) 上述した実施形態では、ダウンロードされた JavaA P の実行が指示された場合に、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成するようにしたが、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトが生成されるタイミングは、JavaA P の実行が指示されたときに限定されるものではない。例えば、JavaA P がデータを参照するタイミングでオブジェ

クトを生成してもよい。

【0090】

(5) 上述した実施形態において、コンテンツサーバ10は、インターネット20に接続されている構成とした。しかしながら、コンテンツサーバ10は、専用線を介して移動パケット通信網30のゲートウェイサーバ31に直接接続されている構成であってもよい。

【0091】

(6) 上述した実施形態では、図15においてハッチングで示すように、KVMと、コンフィグレーションとしてCLDCを備えるとともにプロファイルとしてオリジナルJava拡張プロファイルを備えるJ2MEとが記憶された携帯電話機40に本発明を適用した場合について説明した。しかしながら、Java実行環境は、上述したKVMとJ2MEの組み合わせに限定されるものではない。また、本発明が適用可能な通信装置は、携帯電話機に限定されるものではない。

例えば、同図に示すように、J2MEのプロファイルとして、オリジナルJava拡張プロファイルの代わりにMIDP (Mobile Information Device Profile) を有する構成であってもよい。また、KVMの代わりにJVMを有し、J2MEのコンフィグレーションとしてCLDCの代わりにCDC (Connected Device Configuration) を、また、J2MEのプロファイルとして、例えば、液晶付電話機用プロファイル、TV用プロファイル、カーナビゲーション用プロファイルなどを有する構成であってもよい。さらには、Hot Spotと、J2SE (Java 2 Standard Edition) またはJ2EE (Java 2 Enterprise Edition) とを有する構成であってもよい。

【0092】

(7) また、以上説明したJava実行環境の変形例から明らかなように、本発明は、例えば、PHS (Personal Handyphone System: 登録商標) 端末やPDA (Personal Digital Assistant)、カーナビゲーション装置、パーソナルコンピュータなどの、通信機能を有する各種電子機器に適用可能である。また、本発明は、移動パケット通信網30に収容される通信装置に限定されるものではない。例えば、図16に示すような通信システム2において、LAN50内に設けられたパ

ーソナルコンピュータ 7 0 A ～ 7 0 C に本発明を適用することもできる。

【 0 0 9 3 】

(8) また、上述した実施形態では、Java プログラミング言語により記述された Java A P を用いた場合について説明したが、プログラミング言語は Java に限定されるものではない。例えば、システムに要求されるセキュリティのレベルによっては、C ++ を使用してシステムを構築することも可能である。

【 0 0 9 4 】

【発明の効果】

以上説明したように、本発明によれば、ダウンロードされるプログラムに化体した信頼のレベルとデータの重要度に応じて、通信装置に記憶されているデータに対するアクセス制御を行うことができるため、データに対するセキュリティを確保しながら様々なプログラムによってデータを操作することができる。

【図面の簡単な説明】

【図 1】 本発明の第 1 実施形態に係る通信システムの構成を示すブロック図である。

【図 2】 同実施形態に係る携帯電話機のハードウェア構成を示すブロック図である。

【図 3】 同実施形態に係る携帯電話機において、不揮発性メモリに記憶されている型 A D F ファイルのデータ構成を示す図である。

【図 4】 同実施形態に係る携帯電話機において、Java A P の実行環境を説明するための図である。

【図 5】 同実施形態に係る携帯電話機において、カプセル化オブジェクトを説明するための模式図である。

【図 6】 同実施形態に係る携帯電話機において、非完全カプセル化オブジェクトについて例示する模式図である。

【図 7】 同実施形態に係る携帯電話機において、完全カプセル化オブジェクトについて例示する模式図である。

【図 8】 同実施形態に係る携帯電話機において、C P U により実行されるオブジェクト生成処理の動作を説明するフローチャートである。

【図 9】 同実施形態に係る携帯電話機において、CPUにより実行されるアクセス管理処理の動作を説明するフローチャートである。

【図 10】 同実施形態に係る携帯電話機において、CPUにより実行されるJava A P 終了処理の動作を説明するフローチャートである。

【図 11】 本発明の第 2 実施形態に係る携帯電話機において、不揮発性メモリに記憶されている重要度テーブルのデータ構成を示す図である。

【図 12】 同実施形態に係る携帯電話機において、不揮発性メモリに記憶されているアプリデータ関係テーブルのデータ構成を示す図である。

【図 13】 同実施形態に係る携帯電話機において、CPUにより実行されるオブジェクト使用管理処理の動作を説明するフローチャートである。

【図 14】 本発明の変形例に係る携帯電話機において、不揮発性メモリに記憶されているアプリデータ関係テーブルのデータ構成を示す図である。

【図 15】 本発明の変形例に係り、Java実行環境の変形例を説明するための図である。

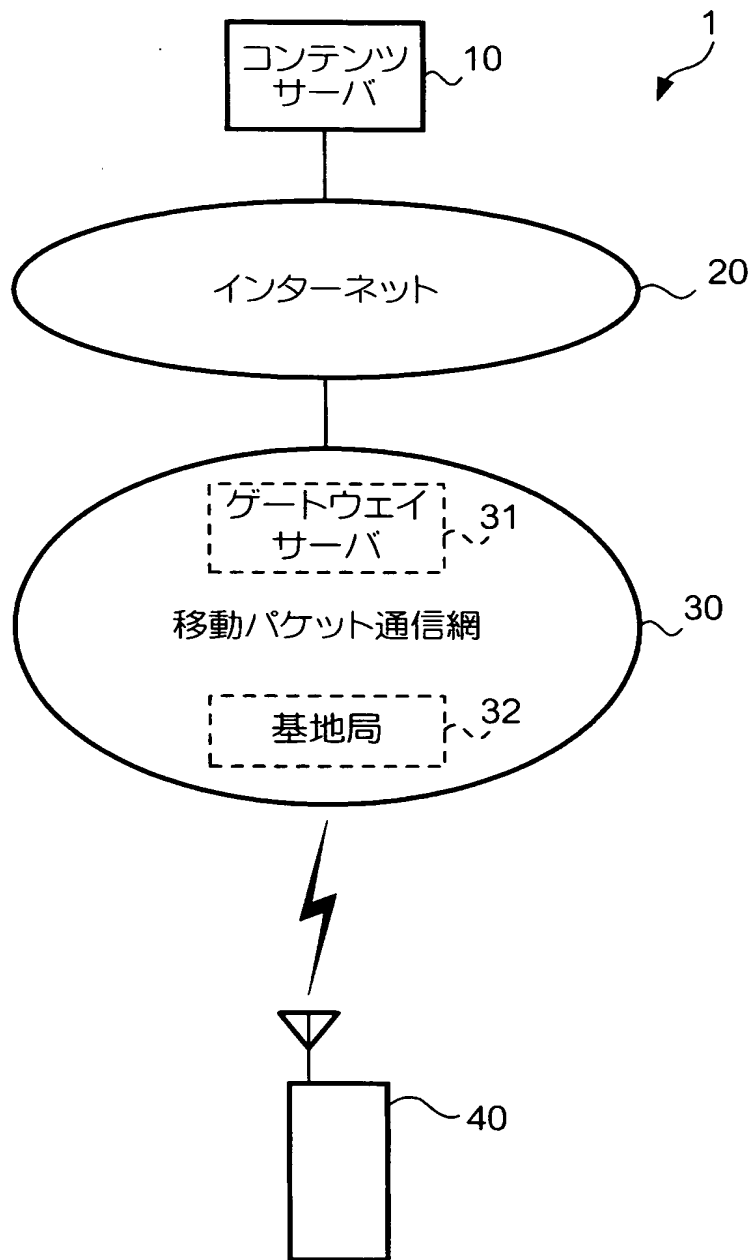
【図 16】 本発明の変形例に係り、通信システムの変形例を例示するブロック図である。

【符号の説明】

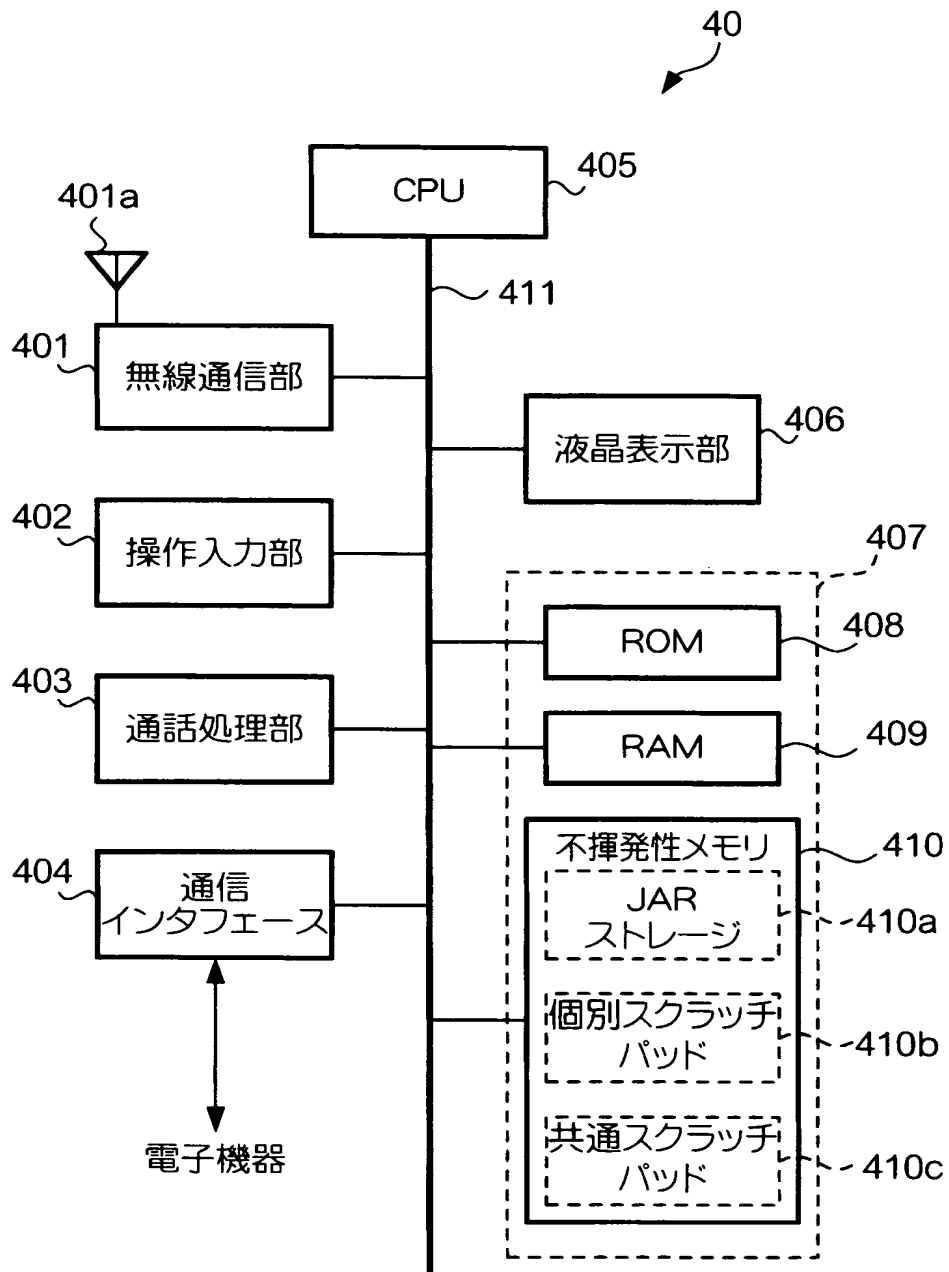
1, 2……通信システム、10……コンテンツサーバ、20……インターネット、30……移動パケット通信網、31……ゲートウェイサーバ、32……基地局、40……携帯電話機、50……LAN、60……ゲートウェイサーバ、70a, 70b, 70c……パーソナルコンピュータ、401……無線通信部、401a……アンテナ、402……操作入力部、403……通話処理部、404……通信インタフェース、405……CPU、406……液晶表示部、407……記憶部、408……ROM、409……RAM、410……不揮発性メモリ、410a……JARストレージ、410b……個別スクラッチパッド、410c……共通スクラッチパッド、410d……重要度テーブル、410e……アプリデータ関係テーブル。

【書類名】 図面

【図 1】



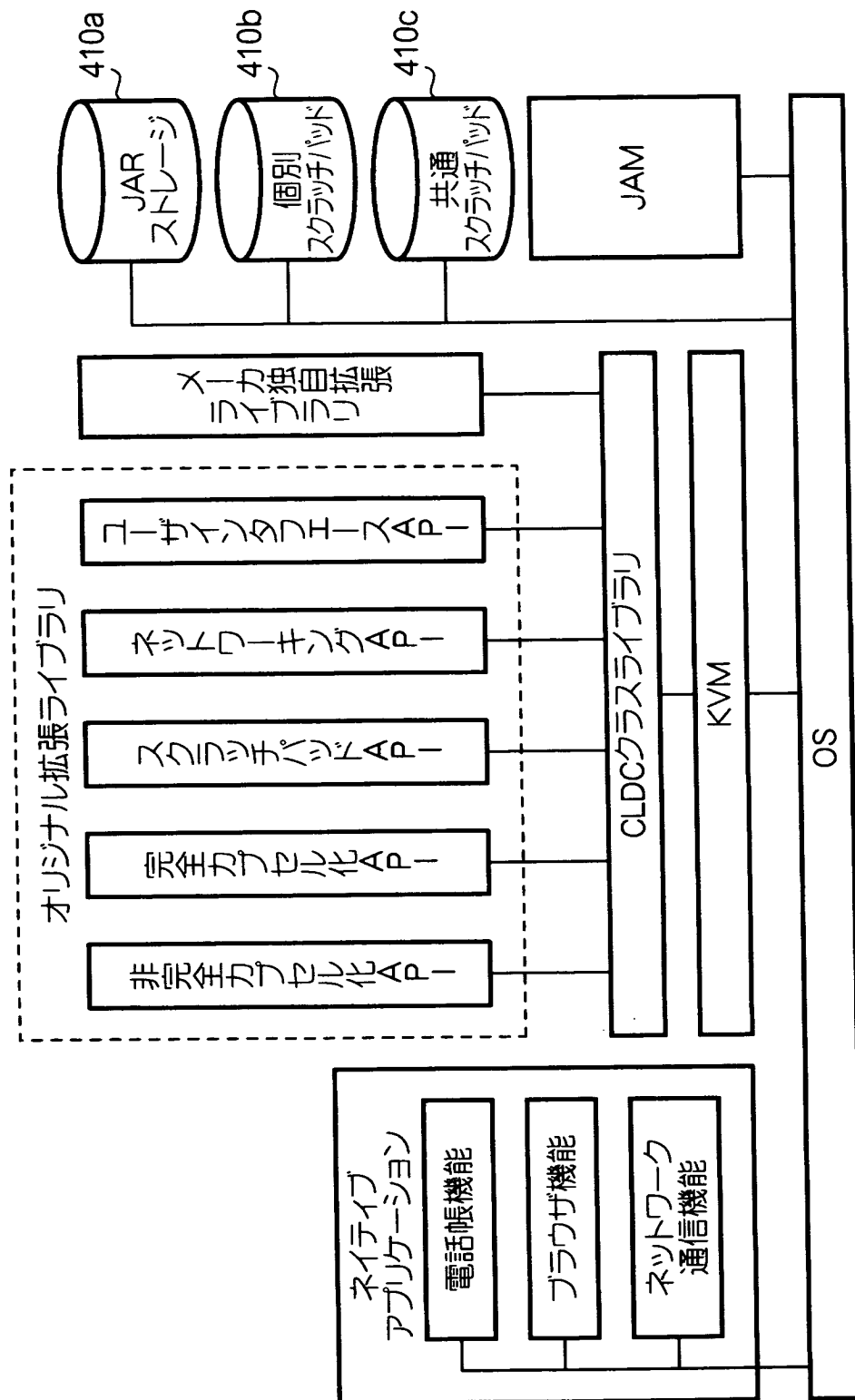
【図 2】



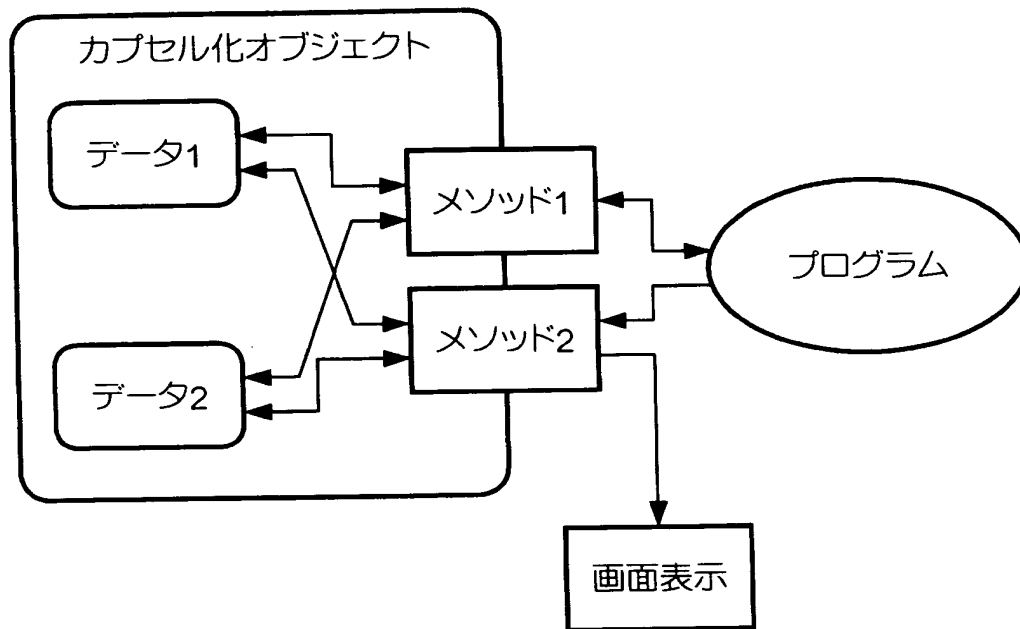
【図 3】

| | | | | | |
|---------|----------------|---------|------|--------------|------------------|
| AppName | Package URL | AppSize | | LastModified | トラステッド アプリ識別子 |
|---------|----------------|---------|------|--------------|------------------|

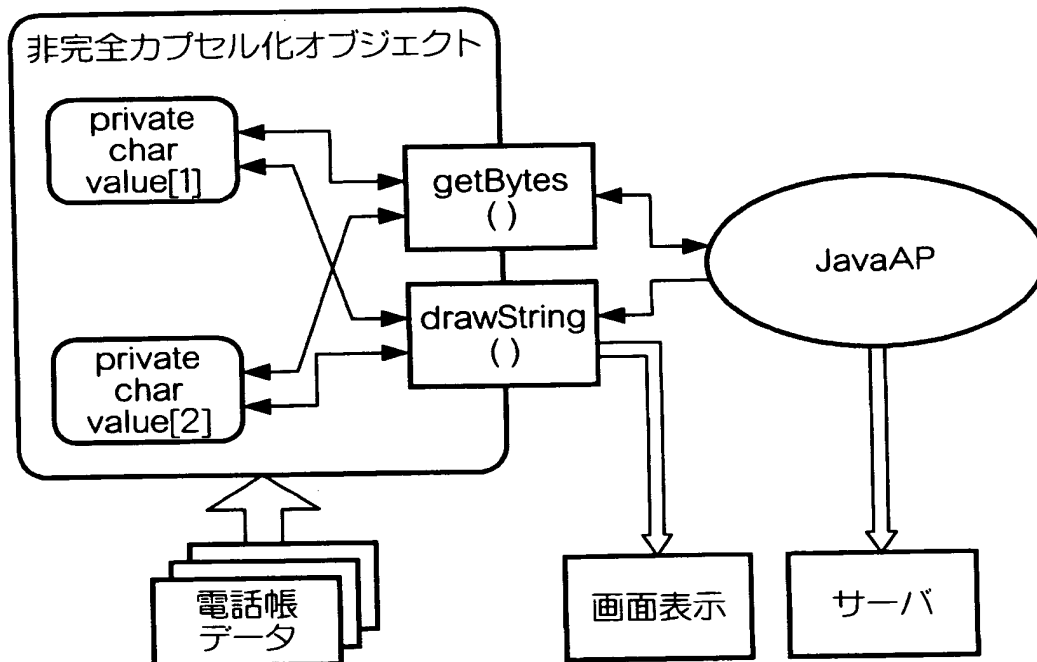
【図 4】



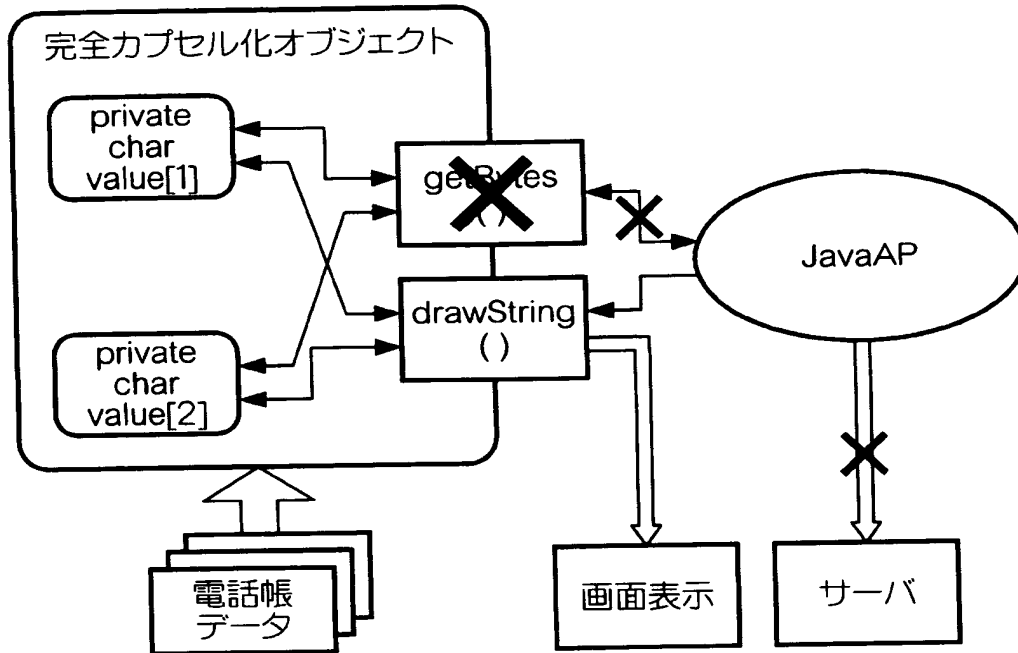
【図5】



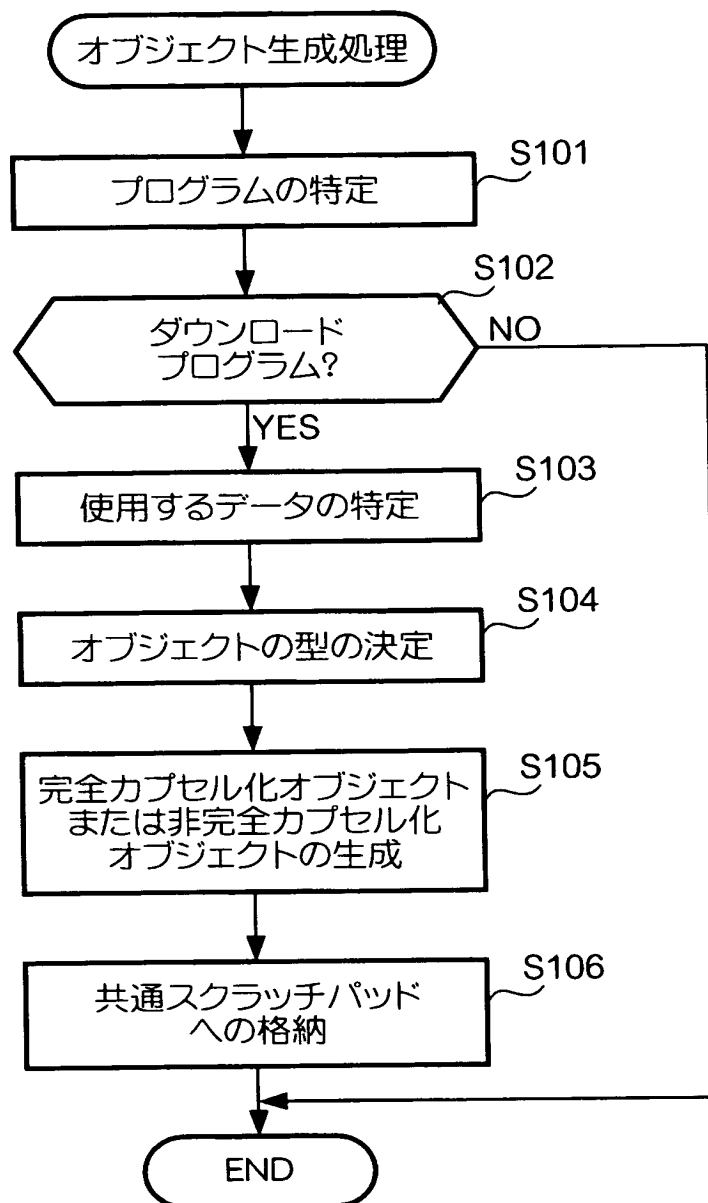
【図6】



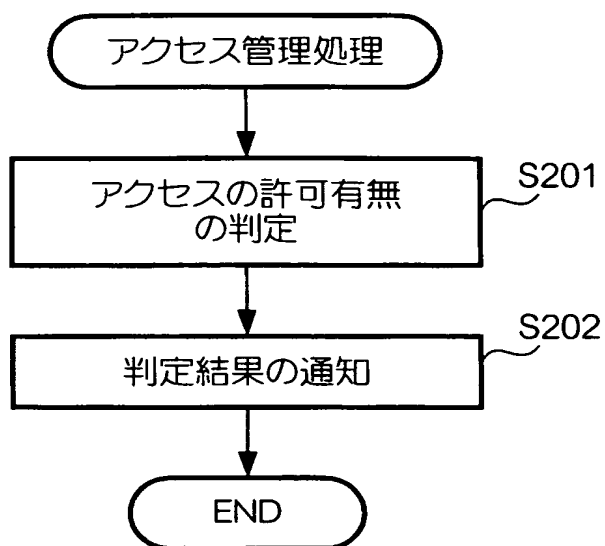
【図 7】



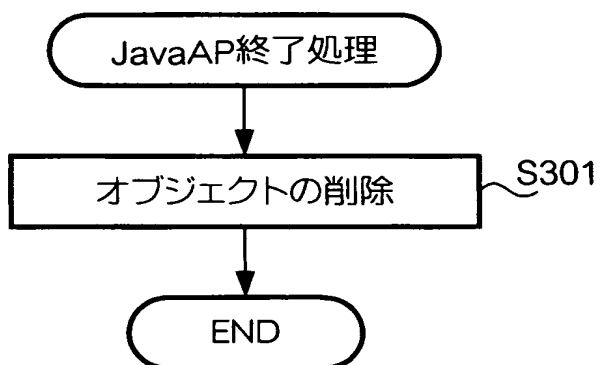
【図 8】



【図 9】



【図 10】



【図 11】

410d

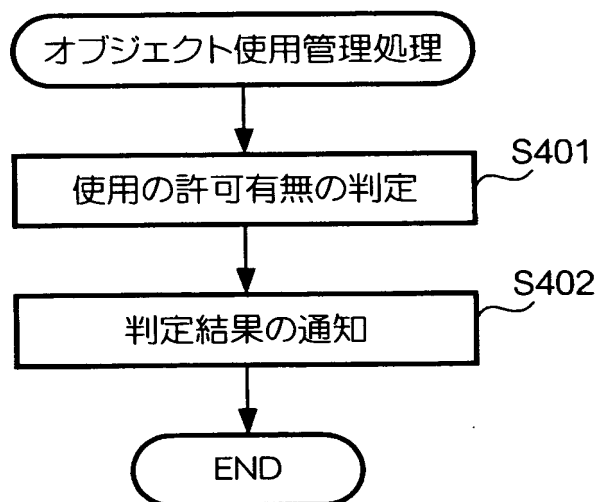
| データ名 | 重要度 |
|------------|-------|
| アドレス帳データ | 高 |
| 電子メールデータ | 高 |
| 着信・発信履歴データ | 高 |
| ユーザデータ | 高 |
| コンテンツA | 中 |
| コンテンツB | 低 |
| 自作画像データ | 低 |
| | |

【図 12】

410e
S

| | | トラステッドアプリ識別子 | | |
|-----------------|----------|--|--|--|
| | | 信頼度 低 | 信頼度 中 | 信頼度 高 |
| データ | 重要度 高 | 完全 カプセル化型 | 完全 カプセル化型 | 非完全 カプセル化型 |
| | 重要度 中 | 完全 カプセル化型 | 非完全 カプセル化型 | 非完全 カプセル化型 |
| | 重要度 低 | 非完全 カプセル化型 | 非完全 カプセル化型 | 非完全 カプセル化型 |
| 使用可能な オブジェクト | | 完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト | 完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト | 完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト |

【図 13】

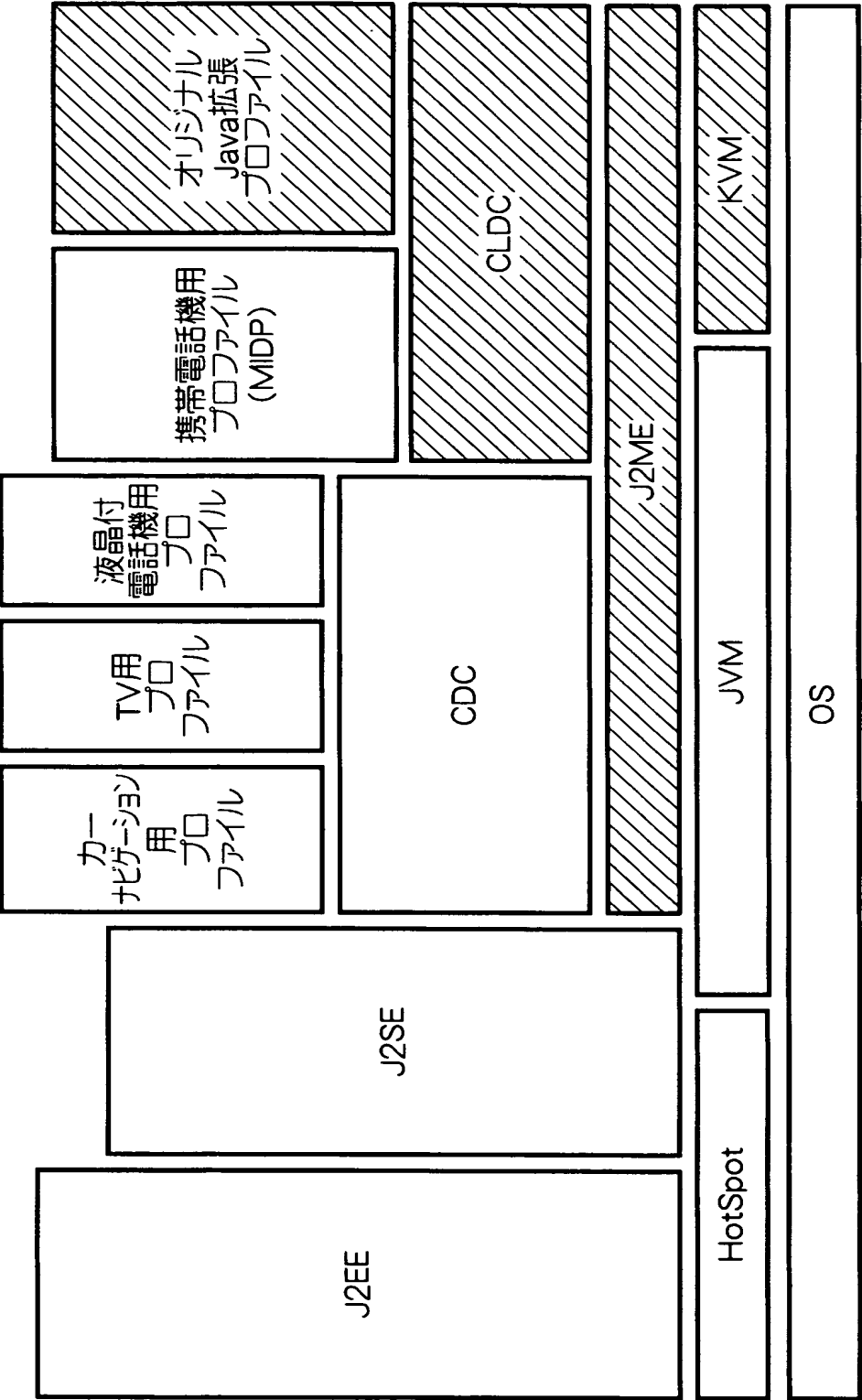


【図 14】

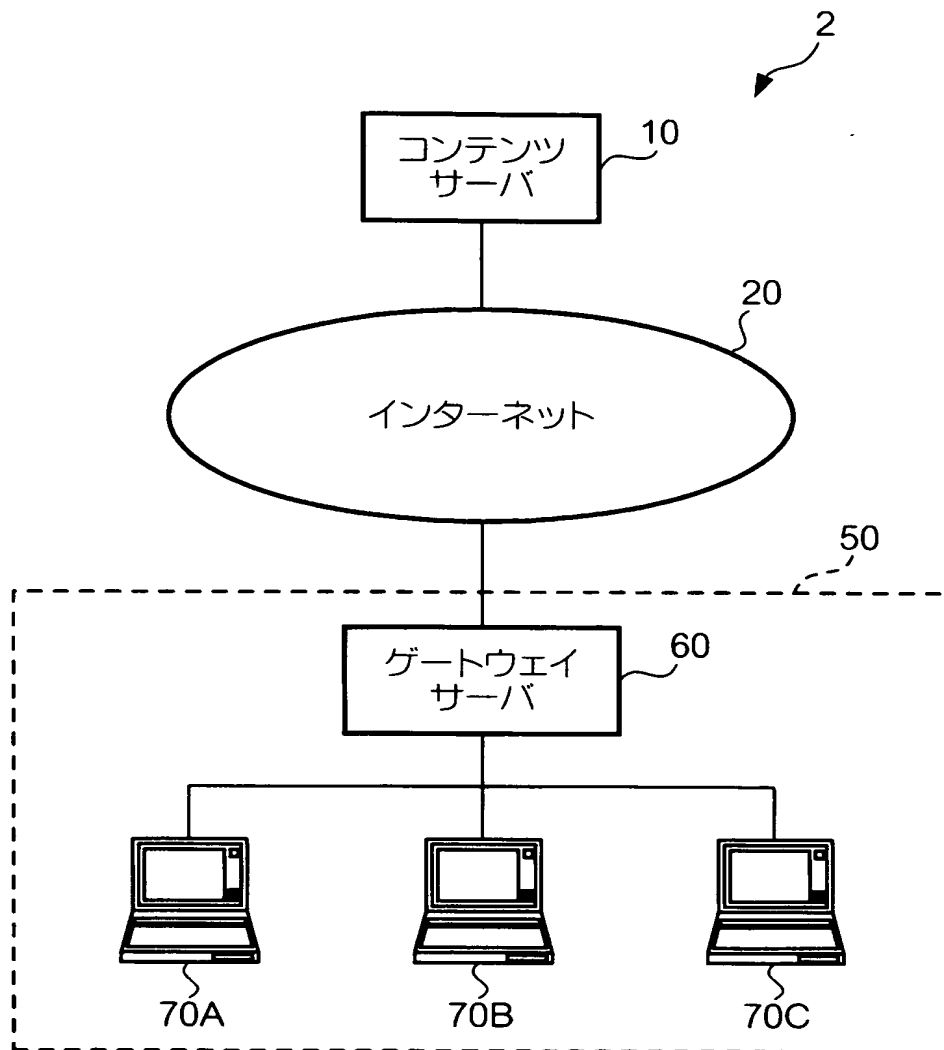
410e

| | | トラステッドアプリ識別子 | | |
|-----------------|----------|---------------|-------------------|--|
| | | 信頼度 低 | 信頼度 中 | 信頼度 高 |
| データ | 重要度 高 | 完全 カプセル化型 | 完全 カプセル化型 | 完全 カプセル化型 |
| | 重要度 中 | 非完全 カプセル化型 | 非完全 カプセル化型 | 非完全 カプセル化型 |
| | 重要度 低 | 非完全 カプセル化型 | 非完全 カプセル化型 | 非完全 カプセル化型 |
| 使用可能な オブジェクト | | なし | 完全カプセル化 オブジェクト | 完全カプセル化 オブジェクト、 非完全カプセル 化オブジェクト |

【図 15】



【図 16】



【書類名】 要約書

【要約】

【課題】 通信装置に記憶されているデータのセキュリティを確保しつつ、当該データを使用する様々なプログラムを提供できるようにする技術を提供する。

【解決手段】 携帯電話機は、プログラムと該プログラムのトラステッドアプリケーション識別情報を受信し、前記プログラムを実行した場合に使用されるデータを特定し、前記トラステッドアプリケーション識別情報に基づいて、生成するオブジェクトの型（非完全カプセル化型、または完全カプセル化型）を選択して選択された型のオブジェクトを生成し、前記プログラムを実行するときには、前記生成されたオブジェクトのみを使用して前記データを使用する。

【選択図】 図 7

特願 2 0 0 2 - 3 1 6 6 3 5

出 願 人 履 歴 情 報

識別番号

[3 9 2 0 2 6 6 9 3]

1 . 変更年月日
[変更理由]

2 0 0 0 年 5 月 1 9 日

名称変更

住所変更

住 所
氏 名

東京都千代田区永田町二丁目 1 1 番 1 号
株式会社エヌ・ティ・ティ・ドコモ